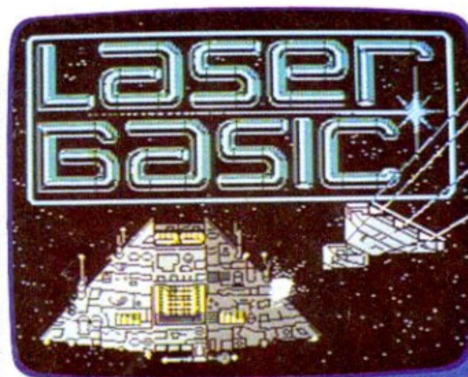
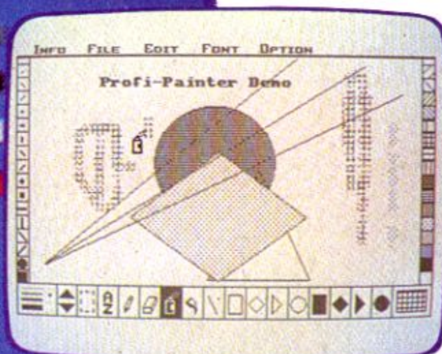
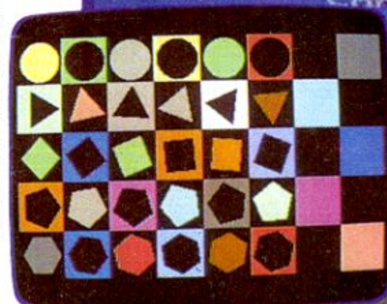
Tutto COMMODORE Grafica

Anno II - Numero 10 - FEBBRAIO 1988 - L. 13.000



COLOUR 1
COLOUR 2
COLOUR 3
BLOCK COL
CHARACTER NO : 078
BLOCK NO : 000





CINQUE ECCEZIONALI PROGRAMMI GRAFICI

- Game Construction Super Kit
- Laser Basic
- Laser Compiler
- Laser Sprite
- Professional Painter



E sei subito artista

Profi-Painter è, come dice il nome stesso, un disegnatore grafico veramente professionale, nell'uso (menù a discesa, icone, help, eccetera) e nei risultati. Le schermate possono poi essere facilmente riutilizzate grazie all'opzione di stampa su disco.

Profi-Painter è un eccellente editor grafico per il C64, infatti permette di sfruttare al massimo la risoluzione grafica del vostro computer, grazie a numerosissime opzioni di lavoro.

Anche se l'uso di questo programma è molto intuitivo, in quanto si viene aiutati dai menù a discesa e dalla richiesta di opzioni tramite icone, riteniamo utile una dettagliata spiegazione sull'uso di Profi-Painter.

Potete caricare il programma direttamente dal menù di *TuttoCommodore* oppure semplicemente digitando LOAD"NOME PROGRAMMA",8,1. Al termine del caricamento potete anche togliere il dischetto *TuttoCommodore* dal drive per inserire un vostro disco dati, tuttavia ricordate che se volete servirvi del primo menù (Info) dovete necessariamente reinserire il disco programma per permettere il caricamento dei file di Help.

Per salvare i vostri lavori è consigliabile che vi serviate di un disco nuovo, interamente dedicato ai lavori PP. A questo scopo dovete prima di tutto formattare un disco; questa operazione può essere effettuata anche tramite PP utilizzando l'opzione Comandi DOS che vedremo in seguito.

I menù a discesa

PP è fornito di cinque menù a discesa. Inserite dapprima il joystick in porta 2. Quasi tutti i comandi di PP saranno in seguito impartiti tramite il joystick. Come potete verificare facilmente muovendo il joystick nelle otto direzioni consentite il puntatore, che all'inizio è al centro dello schermo, si muove dalla stessa parte. Per utilizzare un menù dovrete portare il puntatore nella parte superiore dello schermo, sopra la scritta che raffigura il menù di cui desiderate servirvi, poi premere il tasto Fire. In questo modo comparirà il menù a discesa vero e proprio. Mantenendo premuto il tasto Fire muovete il joystick in alto o in basso finché l'opzione desiderata non appare in reverse. Rilasciando a questo punto il tasto Fire l'opzione è attivata.

• **Info** - Il menù Info serve per ottenere due schermate di aiuto: la schermata Icone mostra il significato delle varie icone che compaiono ai lati dello schermo, mentre la schermata Tasti mostra alcuni significati dei tasti Shift e Commodore. Ricordate che se selezionate una di queste opzioni il disco programma deve essere inserito nel drive.

• **File** - Il menù File serve per tutte le operazioni che coinvolgono il drive. Con il comando Load potete caricare un disegno salvato in precedenza (ovviamente il computer vi chiede il nome del file: se sul disco non esiste un file con quel nome vi viene segnalato un errore). Con il comando Save potete salvare un disegno cui sia già stato assegnato un nome, ad esempio perché l'avete caricato in precedenza, oppure perché l'avevate già salvato con il comando "Nome e Save". Quest'ultimo comando serve per l'appunto ad attribuire un nome a un disegno per poi registrarlo. Se selezionate il comando Stampa potete registrare su disco la bit map della parte di disegno visibile sullo schermo. Per fare questo selezionate "no" e attribuite un nome allo schermo. I file che rappresentano un disegno hanno lunghezza variabile, a seconda della complessità del disegno, e si riconoscono perché il nome termina con il suffisso .DIS. I file che contengono una bit map invece hanno come suffisso .HRD e sono lunghi esattamente 8000 byte, corrispondenti a 32 blocchi.

Per stampare la bit map che avete registrato dovete spegnere il computer e caricare il programma Stampa che si trova nel disco programma. Date il RUN e attendete che compaia il menù. Premendo il tasto A potete caricare la bit map che avete salvato in precedenza (a questo scopo date il nome per esteso, per esempio DISEGNO.HRD). Premendo il tasto B potete rivedere il disegno, mentre con il tasto E potete procedere con la stampa.

L'opzione Catalog vi permette di vedere la directory del disco contenuto nel drive. Se è lunga e non ci sta interamente nello schermo, premendo Fire potete vederne il seguito. Per tornare al lavoro normale selezionate l'opzione OK.

La successiva opzione, Comandi DOS, serve per impartire un qualunque ordine al floppy drive. Se per esempio volete formattare un disco potete selezionare questa opzione e poi digitare N:NOME,ID, mentre se volete cancellare un file potete scrivere S:NOME.

Fate molta attenzione all'operazione seguente: il comando Reset effettua il reset completo del computer, il che significa che se selezionate questa opzione il programma sparirà dalla memoria e con esso il disegno presente sullo schermo a meno che l'abbiate in precedenza salvato.

• **Edit** - Il menù Edit serve per tutte le operazioni di editing di schermo. Vediamo una per una tutte le operazioni di questo menù.

- *Oops!:* serve qualora abbiate commesso qualche errore durante un'operazione di editing. Selezionando Oops! infatti il disegno ritorna com'era prima che venisse eseguita l'ultima operazione.

- *Cancellare:* serve per cancellare completamente il disegno in memoria.

Tutte le altre operazioni di editing possono essere eseguite se avete in precedenza selezionato una parte del disegno con l'ap-

posito evidenziatore che si trova tra le icone e del quale parleremo in seguito. Supponiamo che la selezione sia stata effettuata.

- *Reverse:* scambia i punti bianchi con quelli neri e viceversa, ovvero esegue il reverse della parte selezionata.

- *Ingrandire* e *Ridurre*: si spiegano da sole.

- *Flip oriz* e *Flip vert*: rovesciano il riquadro selezionato (ovviamente in senso orizzontale o verticale).

- *Ruotare:* ruota la parte scelta di novanta gradi.

- *Togliere:* trasferisce su disco la parte selezionata, cancellandola dallo schermo (naturalmente il computer vi chiederà il nome da assegnare al file con il quale il ritaglio verrà registrato).

- *Copiare:* esegue allo stesso modo il salvataggio su disco del Ritaglio, ma non lo cancella dallo schermo.

- *Inserire:* serve per ricaricare da disco un Ritaglio salvato in precedenza. Per utilizzare questa opzione non dovete avere selezionato un dettaglio sullo schermo, bensì procedete nel seguente modo: selezionate l'evidenziatore, portate il cursore sul disegno, premete Fire, selezionate l'opzione Inserire, date al computer il nome con il quale avevate salvato il dettaglio. Ricordate a questo proposito che i file di tipo Ritaglio vengono salvati con il suffisso .RIT-.

• **Font** - Questo menù serve per selezionare il font, ovvero il tipo di carattere con cui procedere per scrivere le parti di testo. Potete scegliere se usare i caratteri standard, oppure di tipo bold (grassetto), oppure i caratteri in corsivo o sottolineati. È inoltre possibile selezionare diverse forme di carattere, nonché crearne nuove (vedi istruzioni seguenti).

• **Option** - Quest'ultimo menù serve per alcune particolari operazioni: con l'istruzione Dis.intero, ad esempio, è possibile vedere sullo schermo l'intero disegno che state realizzando, che a dimensione naturale è grande circa il doppio della parte visibile. Naturalmente per vedere l'intero disegno lo si deve rimpicciolire, perdendo i particolari. Selezionando con il joystick la scritta OK si torna alle dimensioni normali.

L'opzione Zoom serve per ingrandire un particolare del disegno, in modo da lavorare con precisione. L'istruzione Scelta colore permette di modificare il colore del bordo, del fondo e del disegno a proprio piacimento servendosi dei tasti funzione.

PP è dotato di una serie di Retini, del cui uso parleremo in seguito, a proposito delle operazioni di Fill. Potete vedere i retini disponibili nella colonna a destra dello schermo. L'opzione Ridef.retini serve per modificare a vostro piacimento tutti i retini che volete. Selezionandola vi viene chiesto se volete caricare da disco un retino registrato in precedenza, oppure se volete crearne uno nuovo. Ricordate che i file con i quali vengono registrati i retini terminano con il suffisso .TRT-.

Per crearne uno nuovo è sufficiente, dopo avere scelto nella colonna di destra quale modificare, agire con il joystick nel quadrato che compare a sinistra. Premendo il tasto Fire su un pixel bianco lo fate diventare nero e viceversa. Al termine potete scegliere se registrare su disco il nuovo retino, nel qual caso dovete fornirgli un nome. Nel quadrato di destra potete osservare l'effetto complessivo creato dal retino. L'ultima opzione serve per ridefinire il set di caratteri. Essa è essenzialmente identica alla ridefinizione di retini, tranne per il fatto che per scegliere quale carattere modificare dovete premere il tasto corrispondente sulla tastiera del C64. I file che contengono un set di caratteri terminano con il suffisso .CAR-.

Le icone di Profi Painter

Come avete potuto vedere caricando il programma i bordi laterali e quello inferiore dello schermo di PP raffigurano dei simboli, detti icone. A ciascuno di questi simboli corrisponde una particolare operazione.

• **La linea inferiore.** Nell'angolo in basso a sinistra si trovano quattro tipi di tratto. Per selezionarli portateci sopra il puntatore, premete Fire e muovete il joystick in alto o in basso, sempre mantenendo premuto il tasto Fire, finché il puntino non compare accanto al tratto desiderato.

La prima icona, che raffigura due frecce, serve per fare scrollare lo schermo, infatti come già accennato, un disegno di PP è grande circa il doppio della parte visibile. Se provate a porre il puntatore su questa icona, premere Fire e muovere il joystick in su o in giù mantenendo premuto il tasto Fire, potrete vedere il disegno muoversi verso l'alto o verso il basso. Se nell'effettuare questa operazione mantenete premuto il tasto Shift potrete passare direttamente alla parte superiore o inferiore del disegno.

La seconda icona è il cosiddetto Evidenziatore o Selezionatore. Premendo Fire sopra questa icona potete disegnare un rettangolo sullo schermo che costituisce il dettaglio sul quale agiscono tutte le istruzioni di editing viste in precedenza.

La terza icona serve per scrivere sullo schermo. È sufficiente selezionarla (portandovi sopra il cursore e premendo Fire), portare il cursore nel posto dove volete iniziare la scritta, premere ancora Fire e procedere a scrivere quello che si vuole. Questa icona, che raffigura una matita, è la più importante, in quanto serve per disegnare a mano libera.

Potete muovere il cursore per lo schermo e disegnare a piacimento premendo il tasto Fire; premendo il tasto Shift durante questa operazione passate al modo Zoom. Se durante il modo Zoom premete ancora Shift potete muovere la zona ingrandita. Per uscire dal modo Zoom portate il cursore sul rettangolino in alto a sinistra e premete Fire.

L'icona Gomma serve naturalmente per cancellare una parte del disegno, mentre l'icona Spray serve per spruzzare in modo casuale il retino selezionato sullo schermo. L'icona seguente è il cosiddetto Fill che serve per riempire con il retino selezionato una zona dello schermo circoscritta da una linea chiusa. Per utilizzare Fill è sufficiente selezionarne l'icona, portare il puntatore in mezzo alla zona desiderata e premere Fire.

L'icona Riga serve per tracciare linee rette. Il funzionamento è molto semplice, basta posizionare il cursore a un estremo del segmento desiderato, premere Fire e mantenendolo premuto portare il cursore all'altro estremo. Rilasciando il tasto Fire si ottiene il segmento desiderato.

Le altre icone che si trovano nella parte inferiore dello schermo servono per disegnare le figure geometriche che rappresentano. Il loro uso è pressoché identico all'icona Riga.

• **La colonna a destra.** Portando il cursore sulla colonna di destra e premendo Fire potete selezionare il retino desiderato da utilizzare per le operazioni di Fill, Spray e per i Pennelli.

• **La colonna a sinistra.** Nella colonna a sinistra si trovano tutti i Pennelli disponibili. Selezionate dapprima un retino, poi un pennello a vostra scelta. Provate poi a disegnare come se aveste in mano la matita e ammirate il risultato.

Durante l'uso dei pennelli e dello spray potete, premendo il tasto Shift, modificare la velocità di tracciamento.

Fabrizio Pintani

Che bello farsi un videogame!

Potrete costruire i videogame che preferite, che nulla hanno da invidiare a quelli americani, senza dover programmare né in linguaggio macchina né in Basic, ma utilizzando solo i menù dei vari editor: grafica, musica, sprite, effetti sonori e speciali scrolling.

Questo eccezionale programma è in grado di creare qualsiasi gioco di animazione con schermo fisso (il paesaggio è fermo) o con scroll (verticale a 2 velocità), nel quale possono giocare anche contemporaneamente 1 o 2 persone. Si possono inoltre creare sprite, presentazioni, fissare punteggi e bonus.

Per gestire i vari menù e i vari editor è necessario usare il joystick e la tastiera. Nei menù appaiono sulla destra delle lettere: è possibile selezionare l'opzione corrispondente sia attraverso il tasto sia premendo il pulsante del joystick una volta indicata l'opzione con la freccia.

Come si utilizza in pratica

A programma avviato appare sullo schermo il menù principale (**tavola 1**).

Nell'usare il programma si consiglia di seguire la successione degli edit e solo alla fine apportare le modifiche necessarie.

• **Edit Sprite.** Il primo che troviamo è Edit sprite (**tavola 2**). Si seleziona col joystick o si batte la S sulla tastiera. Questo editor serve a disegnare tutti gli sprite del gioco.

- *F1:* Si sceglie lo sprite che si vuole disegnare o variare (ce ne sono127 a disposizione). Col joy si seleziona il numero e schiacciando il pulsante si va in Edit sprite mode (equivale a premere F3), a questo punto sulla griglia (16*21) si può disegnare lo sprite.
- *F5:* Consente di selezionare il colore tra i 16 a disposizione.
- *F7:* Ci sono tre colori costanti per tutti gli sprite (background 1; generalcol2: generalcol3) e uno variabile per ogni sprite (changeable4). Con F7 si sceglie quale variare.
- *S:* Fa slittare lo sprite prescelto nelle direzioni del joystick.
- *M* (Mirror effect): Inverte il disegno in tutte le direzioni.
- *C:* Duplica lo sprite in un'altra posizione. Va prima impostato il numero dello sprite che si vuole copiare quindi si preme Fire, poi va impostato il numero nel quale si vuole mettere la copia dello sprite e si preme nuovamente Fire. Si ottengono così due copie uguali ma con numeri diversi.
- *E:* Cancella lo sprite selezionato.
- *X:* Riporta al menù principale.

Quando si sono costruiti tutti gli sprite se ne seleziona uno con Edit sprite e si passa all'edit seguente.

• **Edit object.** In questa fase (**tavola 3**) si dispone dei seguenti comandi:

- *F1:* Attribuisce allo sprite selezionato una determinata fun-

Tavola 1. Menù principale.

Edit sprite	S
Edit object	O
Edit background	B
Edit sfx	F
Edit player limitations	P
Edit attack waves	A
Edit levels	L
Edit front end	G
Test Game	T
Storage	D

Tavola 2. Edit Sprite

Select sprite	F1
Edit sprite	F3
Edit colour	F5
Select colour	F7
Slide sprite	S
Mirror sprite	M
Copy sprite	C
Erase sprite	E
Exit	X

zione tra le seguenti possibili: giocatore 1 o 2, sparo 1 o 2, esplosione 1 o 2, uno degli 8 spari, una delle 8 esplosioni e uno dei 36 nemici.

- *F3:* Verifica che la funzione attribuita a uno sprite sia corretta.

- *F5:* Consente di cambiare colore all'obiettivo (sprite) selezionato (equivale a changeable4).

- *F7:* Con questa funzione si creano effetti speciali. Per esempio uno sprite (obiettivo) può muoversi nello schermo con la medesima forma, cambiare a ogni movimento del joystick e poi (lasciando il joystick) ritornare nella forma normale o rimanere nella posizione assunta.

Per esempio se lo sprite è fisso si seleziona dal sottomenù ANIM TYPE (T) e si imposta sequence 1; se lo sprite deve invece essere una successione di 5 sprite diversi si selezionerà sequence 5. Si può selezionare uno sprite e metterlo in una delle 18 caselle a disposizione.

Il programma eseguirà la sequenza partendo dall'alto, da sinistra verso destra.

- *S:* Consente di selezionare la velocità con la quale il programma esegue la successione degli sprite per creare l'obiettivo.

- *T:* Oltre a sequence n (n massimo 18) si può selezionare Directional (identico al comando precedente, solo che appare una griglia che simula i movimenti del joystick) e Directional old (lasciando il joystick lo sprite rimane nell'ultima posizione assunta).

- *E:* È un comando molto importante perché fornisce tutti gli attributi che servono a un obiettivo: velocità, valore in punti, colpi necessari per distruggerlo, tipo di sparo (varie direzioni, random cioè casuale, ecc.), frequenza di sparo, velocità del proiettile, decidere quale suono deve fare, quando spara, quando esplode (ci sono 8+8 effetti a disposizione), come esplode, quale colpo utilizza e via di seguito. Si possono definire le collisioni sia con il nemico sia tra il proiettile giocatore-nemico.

- *C:* Duplica un obiettivo; in totale gli obiettivi sono 58.

- *X:* Riconduce al menù principale.

Tavola 3. Edit Object

Select object	F1
Test object	F3
Edit colour	F5
Select sprite and place	F7
Edit anim speed	S
Edit anim type	T
Edit enemy bits	E
Copy object	C
Exit	X

• Edit Background. A questo punto si passa a Edit background (**tavola 4**), che consente di creare lo scenario sullo sfondo.

- *S F1, F3, F5, F7:* Appaiono tre griglie e un quadratino lampeggiante. La griglia in alto (4*8) ingigantisce l'interno del quadratino lampeggiante (carattere); nel rettangolo in basso si possono inserire tutti i caratteri (253); il quadrato a destra è in pratica un blocco in cui si possono depositare i caratteri.

I blocchi poi verranno messi assieme e uniti costituiranno il paesaggio. Il modo di procedere è analogo a quello per la costruzione degli sprite.

- *S:* Seleziona il blocco da modificare o da utilizzare ex novo. Si possono memorizzare un massimo di 128 blocchi.

- *E:* Dopo aver selezionato il blocco, con questo comando lo si può modificare.

- *M:* Costruisce il sottofondo, dopo aver scelto un blocco lo si deposita nel paesaggio.

- *P:* Mette a disposizione un altro modo per modificare il blocco.

- *B:* Duplica un blocco.

- *C:* Duplica un carattere.

- *X:* Riconduce al menù principale.

Tavola 4. Edit Background

Select char	F1
Edit char	F3
Edit colour	F5
Select colour	F7
Select block	S
Edit map	M
Paint block	P
Copy block	B
Copy char	C
Exit	X

• Edit Sfx. Consente di modellare gli effetti sonori speciali, come esplosioni, spari, scontri e tutto quello che serve al videogame.

- *F1:* Permette di selezionare a che cosa verrà attribuito il suono: sparo, esplosione, partenza, vita extra al giocatore 1 o 2,

otto suoni differenti di esplosioni e altri 8 per gli spari.

- *F3:* Dopo aver identificato la causa del suono con l'opzione F1, si costruisce il suono agendo sulle 8 levette del pannello di comando.

Ogni combinazione degli 8 indicatori definisce un particolare effetto che può essere regolato fino ad ottenere quello desiderato. Possono essere d'aiuto i numerosi effetti già predefiniti.

- *C:* Copia un suono.
- *X:* Riconduce al menù principale.

• **Limitations.** Si attribuiscono le varie funzioni ai giocatori.

- *F1 o F2:* Servono per attivare lo sprite, decidere il numero delle vite, la velocità, i colpi, lo sparo direzionabile, il bonus, ecc. Si può inoltre selezionare l'area in cui il giocatore può muoversi e dove deve ricomparire dopo essere morto.
- *X:* Riconduce al menù principale.

• **Edit Attack Waves.** Dopo aver concluso tutte le fasi precedenti si inseriscono i nemici sullo sfondo.

- *F1:* Per prima cosa si sceglie il nemico, si preme Fire e compare lo sfondo con la parola Rough. Appena scelta la posizione si preme ancora Fire e comparirà la parola Fine e il quadro si muoverà con lentezza ma con precisione.

Schiacciando nuovamente, col joystick si metterà lo sprite nemico nel posto desiderato. Dopo aver schiacciato per l'ultima volta si muoverà lo sprite col joystick e il tragitto verrà memorizzato battendo il tasto F7. Durante il gioco, arrivati in quel punto, quello sprite farà sempre quel tragitto.

- *F3:* La domanda "Link to what?" consente di selezionare lo sprite che deve effettuare il tragitto descritto con l'opzione precedente.
- *F5:* Cancella un nemico. Per vedere se gli sprite sono al posto giusto bisogna far scorrere lo schermo con la parola Fine. ".n.units free:" indica la memoria ancora disponibile per inserire nemici.
- *X :* Riconduce al menù principale.

• **Edit Levels.**

- *F1:* Appare una tabella in cui si possono modificare i 22 livelli a disposizione (scroll, avanzamento con giocatore, quadro fisso), la velocità (0,1 , 2), la durata (0-99 sec.).

Queste modifiche si riflettono sulla difficoltà del gioco e quindi consentono di modellare i livelli.

- *F3:* Dopo aver selezionato e impostato il livello prescelto (es. 01), con F3 si definisce quella parte di sfondo che verrà chiamata livello 01.

Quando il computer ha finito di eseguire il livello corrente prima di andare oltre legge la variabile "end of level": loop=ricomincia da capo e cioè dal livello 01, continue= prosegue normalmente; redraw= ripete l'ultima parte del livello.

Tavola 5. Storage

Load	
Save sprites	
Save background	
Save objects	
Save sfx	
Save attack waves	
Save levels	
Save character set	
Save all data	
Save finished game	
Change device	C
Exit	X

• **Edit Front End.** Serve per creare la presentazione al gioco.

- *F1:* Consente di ridefinire i caratteri.
- *F3:* Scrive il messaggio che dovrà essere visualizzato.
- *F5:* Seleziona i colori e l'effetto visivo.
- *F7:* Riconduce al menù principale.

• **Test Game.**

- *F1:* Consente di provare il funzionamento del gioco dall'inizio.
- *F3:* Consente di provare il gioco partendo da un certo livello. Dopo aver selezionato un livello con F3, il gioco inizia da quel livello.
- *X:* Riconduce al menù principale

• **Storage.** Consente di memorizzare su disco o su nastro parti dei giochi che si stanno assemblando (**tavola 5**) e al termine la versione definitiva e funzionante.
Ovviamente i giochi in versione definitiva girano da soli e non richiedono la presenza del programma di costruzione.

Nota

Da ogni esecuzione si può tornare al sottomenù premendo la barra spaziatrice.

All'inizio le procedure di costruzione delle varie parti potranno risultare un po' macchinose e lente, ma si tratta solo di farci un attimo la mano.

Con un po' di pratica e di prove ci si rende conto che senza le difficoltà della programmazione si possono davvero costruire giochi spettacolari in breve tempo.

Oscar Maeran

Basic? Laser!

Laser Basic è senza dubbio una tra le più potenti espansioni Basic che si possano trovare in commercio. Pur essendo dedicata prevalentemente alla gestione della grafica mette a disposizione anche una serie di validi aiuti per facilitare la gestione degli effetti sonori e il lavoro di stesura e debug dei programmi.

Una volta caricato e lanciato Laser Basic, sul video viene visualizzato un menù con una serie di opzioni per la configurazione dell'assetto di lavoro, e cioè:
1) Turbo tape, multi tasking.
2) No Turbo tape, multi tasking.
3) Turbo tape, no multitasking.
4) No turbo tape, no multitasking.

Ogni opzione viene attivata premendo il numero che la identifica. Ovviamente per chi lavora con il registratore è raccomandato l'impiego del turbo tape che è in grado di velocizzare enormemente le operazioni di input e output su questa periferica. Il multitasking verrà spiegato nei dettagli più avanti; per ora basti sapere che per multitasking si intende l'esecuzione contemporanea di più programmi in memoria. Sotto ogni opzione viene indicato l'ammontare di memoria a disposizione se l'opzione stessa viene attivata.

Le istruzioni messe a disposizione da Laser Basic verranno descritte secondo un ordine logico corrispondente alla funzione delle istruzioni stesse. Prima di tutto è necessario fare alcune osservazione di carattere generale. Con Laser Basic nel comando List è necessario utilizzare il carattere "," (virgola) al posto di "-" (segno meno). Ad esempio List 100-200 con Laser Basic diventa List 100, 200. Inoltre le parole chiave non possono essere più abbreviate con i caratteri shiftati ma utilizzando il carattere "." (punto). La lista completa delle abbreviazioni consentite è riportata nella **tavola 1**. Per finire il comando Rem è sostituito dal carattere "'" (apice singolo).

Programmazione strutturata

Le istruzioni indicate di seguito si aggiungono alle consuete strutture di controllo messe a disposizione dal Basic standard (For-Next e If-Then) e quindi possono essere utilizzate insieme a queste.

• **If-Then-Else:** è una estensione dell'istruzione If-Then. Funziona in questo modo: viene valutata l'espressione booleana (condizione) indicata dopo la parola chiave Then e se questa risulta vera l'esecuzione del programma continua con l'istruzione o le istruzioni indicate dopo la parola chiave Then. Nel caso in cui l'espressione sia falsa l'esecuzione continua con le istruzioni indicate dopo Else. Le tre parole chiave devono stare sulla stessa linea di programma, si tenga inoltre presente che si possono avere più If-Then-Else sulla stessa linea e che ai valori -1 e 0 che corrispondono in basic standard alle condizioni vero e falso sono associate le due costanti predefinite True e False. Ecco un esempio:

If a then print 1 else if b then print 2 else print 3.

L'effetto è il seguente: viene visualizzato 1 se a è true (cioè se a vale -1); viene visualizzato 2 se a è false (cioè se a vale 0) e b è true (cioè b=-1); viene visualizzato 3 se sia a che b sono false (cioè a=b=0). Un'ultima nota: ogni Else viene considerato associato al più recente Then; inoltre se dopo la parola chiave Then c'è un comando o un assegnamento la parola chiave Then può essere omessa.

• **Cif-Celse-Cend:** è la versione potenziata della struttura di controllo precedente. Mentre If-Then-Else deve essere mantenuta su una sola linea di programma, Cif-Celse-Cend può essere estesa a più linee. Il comportamento è il seguente: viene valutata l'espressione booleana specificata dopo la parola chiave Cif e se il valore ottenuto è true (-1) vengono eseguite tutte le linee di programma seguenti Cif fino alla prima linea di programma che contiene Celse e poi il programma prosegue con le linee seguenti la parola chiave Cend. Nel caso in cui l'espressione booleana fornisca il valore false (0) l'escuzione del programma continua alle linee di programma specificate dopo la parola chiave Celse e quindi con la porzione di programma seguente la parola chiave Cend. Anche in questo caso il ramo Celse (equivalente a Else dell'istruzione precedente) può essere omesso. L'effetto è quello che si otterrebbe inserendo delle linee vuote fra Celse e Cend. Ecco un esempio:

```
100 Cif   st<>0
110     open 15,8,15
120     input#15,w$,x$,y$,z$
130     printw$;",";x$;",";y$;",";z$
140     er=1
150     close 15
160 Celse
170     er=0
180 Cend
```

Questa porzione di programma quando viene eseguita verifica lo stato del drive e visualizza il messaggio d'errore del Dos se ci sono stati degli errori. Se non si sono verificati errori la variabile er vale 0.

• **Repeat-Until:** è una struttura di controllo di iterazione e il suo funzionamento è il seguente: vengono eseguite tutte le linee di programma comprese fra la parola chiave Repeat e Until, quindi viene valutata l'espressione booleana indicata dopo Repeat. Se il valore ottenuto è false (0) vengono nuovamente eseguite le linee di programma fra le due parole chiave Repeat e Until mentre se si ottiene il valore true (-1) l'esecuzione del programma continua con la linea di programma che segue Until. Ecco un esempio:

```
10 repeat
20     input"batti 3 caratteri";a$
30 until len(a$)=3
```

Se esguite queste linee creeranno un ciclo finché non verrà inserita una stringa di 3 caratteri.

• **While-Wend:** questa struttura di controllo di iterazione è praticamente simmetrica rispetto alla precedente. Prima di tutto viene valutata l'espressione booleana indicata dopo la parola chiave While. Se questa fornisce il valore true (-1) vengono eseguite tutte le linee di programma fra le parole chiave While e Wend e quindi si ritorna a valutare la condizione. In caso contrario, cioè se la condizione dopo While risulta falsa (0), l'esecuzione del programma continua con le linee che seguono Wend.

• **Case-Of-Casend:** è una struttura di controllo di selezione e permette di selezionare una fra numerose alternative. Facciamo un esempio:

```
10 input a
20 case a
30 of 3:print"hai selezionato il livello
più difficile"
40 of 2:print"hai selezionato il livello
medio"
50 of 1:print"hai selezionato il livello
più semplice"
60 of or:print"questo livello non esiste
!!!"
70 casend
```

Quando questo breve programma viene eseguito vengono visualizzati i messaggi indicati alle linee 30, 40, 50 o 60 a seconda che la variabile a assuma i valori 3, 2, 1 oppure <> da uno di questi tre valori. Dopo la parola chiave Case deve sempre essere indicata una variabile il cui valore verrà utilizzato per selezionare la porzione di programma da eseguire. La parte di programma che verrà eseguita è quella che inizia dalla linea che contiene la parola chiave Of seguita dal valore assunto dalla variabile indicata dopo Case. Ovviamente è possibile specificare più linee di programma per ogni sezione di Case. Se la variabile assume un valore che non trova riscontro in nessuna delle sezioni Case vengono eseguite le linee di programma indicate fra Case Or e Casend.

• **Label:** anche le Label (etichette) rientrano nell'ambito della programmazione strutturata dal momento che sono un valido aiuto per la stesura di programmi facilmente leggibili e modificabili. In generale per label si intende un insieme di caratteri alfanumerici che inizia con una lettera. Anche con Laser Basic le label devono sempre iniziare con una lettera ma al loro interno possono contenere anche i caratteri "$" (dollaro) e "%" (percentuale). Le Label servono per identificare una linea di programma in modo tale che ogni riferimento ad essa possa essere effettuato utilizzando la label stessa piuttosto che il numero di linea. Ecco un esempio:

```
10 Input"Raggio sfera";a
20 Print"v:calcolo volume"
30 Print"s:calcolo superficie"
40 Repeat
50    Get a$
60 Until (a$="s" or a$="v")
70 if a$="s" Gosub superficie
80 Gosub volume
90 Print"risultato"r
100 end
110 Label superficie
120 r=4*pi*(a^2)
130 Return
140 Label volume
150 r=(4*pi*(a^3))/3
160 Return
```

Si noti anche che la costante PI (predefinita) corrisponde al valore di p greca. Ovviamente le label possono essere utilizzate da tutte le istruzioni di salto (Goto, Gosub, On..Gosub, On..Goto, Run, Then) e dalle istruzioni Restore e On..Restore. Queste ultime istruzioni permettono di modificare a piacimento il puntatore alla linea data corrente. Ecco un esempio di utilizzo di queste due istruzioni:

```
10 Restore blocco2
20 Read a:Print a:End
900 Data 5,8,9,0
1000 Label blocco2:Data 7,0,9,9
```

La linea 10 setta il puntatore alla linea data con il valore corrispondente alla linea di programma identificata da blocco2, cioè 1000. Di conseguenza l'istruzione Read leggerà il valore 7 che poi verrà visualizzato.

```
10 Input"numero programma (1, 2 o 3)";a
20 On a Restore gioco, utility, gestionale
30 Read a$
40 Load a$,8,1
50 Label gioco:Data"space force"
60 Label utility:Data"renumber"
70 Label gestionale:Data "conto corrente"
```

La linea 20 setta opportunamente il puntatore alla linea data in relazione al valore della variabile a. Se a vale 1 verrà caricato il programma space force; se vale 2 verrà caricato renumber; se vale 3 verrà caricato conto corrente.

• **Procedure:** Una procedura è in pratica un sottoprogramma. Rispetto a una normale subroutine tuttavia la procedura ha in più la facoltà di permettere il passaggio di parametri e di definire al suo interno delle variabili locali. Ecco un esempio di procedura:

```
100 Label centra (a$,y)
110 local l
120 l=len(a$)
130 print@20-l,y;a$
140 procend
```

Digitando Proc ("prova",10) verrà visualizzata la stringa "prova" centrata sulla decima linea dello schermo. La procedura può essere utilizzata anche in modo programma e non solo in modo diretto. Ogni procedura deve iniziare con l'istruzione Label seguita dal nome della procedura stessa ed eventualmente dalla lista di parametri. I parametri, che devono essere separati tra loro dalla virgola, possono essere in numero qualsiasi, l'importante è che stiano tutti su una linea di programma. Ogni procedura deve terminare con Procend. Per richiamare una procedura basta utilizzare l'istruzione Proc seguita dal nome della procedura e dalla lista di parametri indicata fra parentesi. La variabile locale definita nella procedura è una variabile che può essere utilizzata unicamente all'interno della procedura ed esiste solo all'interno della procedura. Quindi è possibile utilizzare all'interno del programma principale una variabile con lo stesso nome di quella locale poiché le due variabili, pur avendo lo stesso nome, saranno distinte. La procedura vista nell'esempio funziona con parametri costanti. Quindi se nella chiamata di procedura (istruzione Proc) vengono specificate delle variabili il loro valore non verrà alterato dalle istruzioni della procedura. Le due variabili a$ e y (dette parametri formali) sono da intendersi locali alla procedura e in esse vengono copiati i valori delle va-

riabili passate come parametri (tali variabili sono i cosiddetti parametri attuali). Sempre nell'ultimo esempio compare l'istruzione print@. Questa istruzione, la cui sintassi è print@x, y (;stringa), permette di posizionare il cursore alle coordinate x, y (0<=x<=39, 0<=y<=24) e (opzionalmente) di visualizzare una stringa a partire da quella posizione. È possibile fare in modo che una procedura acceda alle variabili passate come parametri e quindi ne modifichi il valore originario che avevano al momento dell'attivazione della procedura. Per specificare che un parametro deve essere variabile, cioè può essere modificato dalle istruzioni della procedura, si deve far precedere il nome del parametro dalla parola Var. Ecco un esempio:

```
10 Label scambia (Var a$, Var b$)
20 local c$
30 c$=a$:a$=b$:b$=c$
40 procend
```

Ora digitando in modo diretto q$="qq":z$="zz":proc scambia (q$,z$) il contenuto delle due variabili q$ e z$ verrà scambiato. Ovviamente all'interno della procedura le variabili globali, cioè quelle definite nel programma principale, sono sempre accessibili se non sono state dichiarate variabili locali con lo stesso nome. È possibile passare anche gli array come parametri. In questo caso è necessario far precedere il parametro all'interno dell'intestazione della procedura dalla parola Var. Inoltre l'array deve sempre essere dimensionato prima di chiamare la procedura, anche nel caso in cui le sue dimensioni sono minori del valore di default (cioè 10).

Laser Basic mette a disposizione una funzione per conoscere esattamente le dimensioni di un array. La funzione in questione è Size e funziona in questo modo: Size (nome-array,0) fornisce come risultato il numero delle dimensioni del vettore indicato come parametro. Supponiamo ad esempio di aver definito un array in questo modo: Dim xt(10, 20). Allora Size(xt,0) ritornerà il valore 2. Con Size (nomearray, n) verrà fornita l'ampiezza dell'n-esima dimensione. Quindi Size (xt, 1) restituirà il valore 21.

• **Funzioni** (su più linee di programma): le funzioni sono simili alle procedure, l'unica differenza consiste nel fatto che ritornano un valore. Per delimitare la fine di una funzione si deve usare il carattere "=" (uguale) seguito dal valore da ritornare. Per chiamare una funzione si deve utilizzare l'istruzione Cnf seguita dal nome della procedura e dai parametri (se presenti nell'intestazione della funzione) fra parentesi. Ecco un esempio:

```
10 Label fattoriale (n)
20 Local i, j
30 i=1
40 Cif n>1
50   For j= 2 to n
60     i=i*j
70   Next j
80 Cend
90 =i
```

Aiuto alla programmazione

• **Exit:** Con questa istruzione è possibile uscire da un loop (ciclo) For..Next, Repeat..Until o While..Wend prima che venga soddisfatta la condizione vera e propria di uscita e senza nessun pericolo di compromettere il corretto funzionamento del programma. Ecco un esempio:

```
10 for t=0 to 100
20 get a$:if a$="" then 20
30 if a$="x" then exit
40 next
```

Alla linea 30 viene testato il carattere letto e se questo è "x" si esce dal ciclo anche se la variabile contatore t non ha ancora raggiunto il valore 100 (condizione di uscita normale). Uscire da un ciclo utilizzando l'istruzione Goto comporta una alterazione spesso distruttiva dello stack.

• **Numeri esadecimali:** Laser Basic permette di specificare numeri esadecimali semplicemente facendo precedere il numero dal carattere "$" (dollaro). I numeri così indicati possono essere liberamente mischiati, all'interno di funzioni e istruzioni, a numeri decimali. È anche possibile convertire un numero da decimale a esadecimale con la funzione Hex$. Ad esempio l'istruzione Print (19456) visualizzerà $4C00. Il numero decimale da convertire deve sempre essere un intero compreso fra 0 e 65535 (estremi inclusi).

• **Deek e Doke:** queste due istruzioni permetto di effettuare un doppio Peek e Poke rispettivamente. Deek è una funzione che fornisce sempre come risultato un intero compreso fra 0 e 65535. Ad esempio Print Deek (4910) corrisponde a print Peek (4910)+256*Peek(4911). Ecco invece come funziona Doke: Doke 4912, $44FF corrisponde a poke 4912, $FF:poke 4913, $44.

• **Dload e Dsave:** La sintassi di queste due istruzioni è la seguente Dload"nomeprogramma" e Dsave"nomeprogramma" e corrispondono rispettivamente a load"nomeprogramma",8 e save"nomeprogramma",8.

• **Pull:** Rimuove due byte dalla cima dello stack. In pratica questa istruzione permette di modificare l'indirizzo di ritorno di un sottoprogramma. Ad esempio se ci si trova in un sottoprogramma e si vuole tornare al livello precedente a quello da cui si è chiamato il sottoprogramma si devono utilizzare le seguenti istruzioni: Pull:Return.

• **Disable:** Disabilita il tasto Run/Stop. Questo comando può essere utilizzato solo in modo programma. Il tasto Run/Stop viene automaticamente riabilitato queando si esce dal modo programma.

• **Dir:** Visualizza la directory. Eventuali programmi in memoria non vengono danneggiati.

• **Old:** Ripristina un programma basic distrutto in seguito all'esecuzione del comando New.

• **Auto:** Consente di attenere la numerazone automatica delle linee. La sintassi è Auto n, dove n è l'incremento fra le linee. Per disabilitare questa funzione si deve digitare Auto e poi premere il tasto Return.

• **Renum:** Effettua la rinumerazione del programma basic in memoria aggiustando anche i riferimenti all'interno delle istruzioni Goto, Gosub e Restore. La sintassi è Renum linea-di-partenza, incremento. Il parametro incremento è facoltativo e se viene omesso l'incremento fra le linee è fissato automaticamente a 10. La rinumerazione non avrà luogo se nel programma ci sono dei riferimenti errati. Ovviamente Laser Basic provvederà a segnalare le linee di programma in cui sono presenti i riferimenti incompatibili.

Grafica

Laser Basic mette a disposizione una serie di istruzioni molto potenti per la gestione di tutti i modi grafici del C64 (bassa risoluzione, alta risoluzione monocromatica e multicolor, extended background) e degli sprite (che d'ora in poi chiameremo hardware sprite). Inoltre consente di utilizzare un nuovo e potentissimo stumento: il software sprite. Per sofware sprite si intende un'area di memoria di grandezza definibile dall'utente che gode di tutte le proprietà della pagina in alta risoluzione. Le dimensioni di uno sprite sono misurate in caratteri cioè in blocchi di 8x8 byte. Ogni sprite - ce ne sono 256 - può raggiungere la dimensione massima di 255x255 caratteri. Per default si hanno a

disposizione 7 K di memoria per memorizzare i dati di tutti i software sprite; tuttavia lo spazio disponibile può essere controllato a piacimento dall'utente e adattato a ogni situazione. Ci sono due tipi di software sprite: Hi-Res sprite e Character sprite. Gli Hi-Res sprite sono pezzi di alta risoluzione utilizzabili sia in modo monocromatico che a colori. I Character sprite sono invece dei segmenti di video in bassa risoluzione e quindi si prestano a contenere dati visualizzabili in uno dei modi grafici utilizzabili in modo testo. Ogni sprite (d'ora in poi con sprite intenderemo software sprite) è identificato da un numero compreso fra 0 e 255.

Ogni sprite può essere o Character o Hi-Res; tuttavia tenete presente che lo sprite 0 corrisponde all'alta risoluzione e quindi deve essere considerato sempre come sprite Hi-Res (di dimensioni pari a 40x25 caratteri) mentre lo sprite 255 è il video è quindi deve sempre essere considerato Character (anch'esso di dimensioni 40x25 caratteri). Inoltre è da sottolineare che lo schermo, cioè lo sprite 255 non occupa memoria all'interno dell'area di 7 K (o delle dimensioni fissate) disponibile per la definizione degli sprite.

Ecco le istruzioni dedicate alla gestione degli sprite:

- **Reset:** Cancella tutti gli sprite presenti in memoria. Non necessita di alcun parametro.
- **Sprite:** Istruzione con cui è possibile definire uno sprite Hi-Res. La sintassi è Sprite numsprite, xcaratteri, ycaratteri. Numsprite è il numero dello sprite che si vuole definire Hi-Res mentre xcaratteri e y caratteri sono rispettivamente le dimensioni, cioè il numero di caratteri, orizzontali e verticale dello sprite. Non è possibile definire uno sprite già esistente inoltre non è possibile definire un nuovo sprite se non c'è più memoria disponibile. Tenete presente che ogni Hi-Res sprite occupa un'area di memoria pari a 7+10*xcaratteri*ycaratteri bytes. Per conoscere l'ammontare di memoria disponibile nell'area riservata per gli sprite si può utilizzare la funzione Sfre. Questa funzione non necessita di alcun parametro.
- **Csprite:** È l'istruzione che permette di definire un sprite character. La sintassi è identica a quella vista per l'istruzione Sprite. Ogni Character sprite occupa un'area di memoria pari a 7+2*xcaratteri*ycaratteri bytes.
- **Wipe:** Rimuove uno sprite, character o hires, dalla memoria. La sintassi è Wipe numsprite, dove numsprite è il numero dello sprite da cancellare. Non è possibile utilizzare questo comando per cancellare lo sprite 0 (cioè la pagina in alta risoluzione vera e propria) o lo sprite 255 (cioè lo schermo).
- **Dfa:** È una funzione con sintassi Dfa (numsprite) e dà l'indirizzo di inizio della zona di memoria che contiene i dati relativi allo sprite numsprite. Ad esempio per sapere dove Laser Basic mette la memoria video digitate: print dfa (255). Dfa restituisce il valore -1 se lo sprite numsprite non è definito.
- **Afa, Afa2:** Queste due funzioni hanno la stessa sintassi della funzione Dfa e ritornano gli indirizzi dell'atributo principale e secondario dello sprite numsprite (per attributo principale si intende il colore di visualizzazione mentre per attributo secondario il colore di sfondo).

Laser Basic mette a disposizione anche una serie di variabili che contengono importanti informazioni sugli sprite. Eccole (accanto al nome viene indicato l'uso):

Spn - numero sprite.
Col - colonna all'interno dello sprite.
Row - riga in uno sprite.
Wid - dimensione orizzontale (in caratteri) dello sprite.
Hgt - dimensione verticale dello sprite (sempre in caratteri).
Spn2 - secondo sprite.
Col2 - colonna all'interno del secondo sprite.
Row2 - riga all'interno del secondo sprite.
Num - quantità di pixel di cui effettuare lo scrolling, lati di un poligono ecc.
Inc - Inclinazione di un poligono in gradi.
Atr - valore corrente dell'attributo principale.
Ccol - colonna all'interno dello sprite in corrispondenza della quale si è verificata una collisione.
Crow - riga all'interno dello sprite in corrispondenza della quale è stata rilevata una collisione.

Tutte le variabili possono esere utilizzate come normali variabili. Le variabili appena viste vengono automaticamente manipolate da Laser Basic durante l'esecuzione di una istruzione. Ad esempio l'istruzione Sprite 10, 2, 4 definisce uno sprite Hi-Res delle dimensioni di 2x4 caratteri. Digitando l'istruzione print Spn, Wid, Hgt verranno visualizzati sullo schermo i valori 10, 2 e 4. Le variabili sprite possono essere utilizzate per rendere più compatti i programmi. Ogni istruzione che agisce sugli sprite può infatti essere utilizzata con parametri oppure senza, se precedentemente sono state settate le variabili sprite necessarie. Ad esempio al posto di Sprite 5, 10, 20 si può scrivere Spn=5:Wid=10:Hgt=20:Sprite.

- **Store:** Permette di salvare su nastro gli sprite in memoria. La sintassi è: Store"nomefile".
- **Dstore:** è l'istruzione che permette di salvare gli sprite su disco. La sintassi è Dstore"nomefile",8.
- **Recall:** Carica da nastro una serie di sprite. Tutti gli sprite presenti in memoria saranno persi. La sintassi dell'istruzione è: Recall"nomefile".
- **Drecall:** Carica da disco una serie di sprite. Anche in questo caso tutti gli sprite presenti in memoria saranno distrutti. La sintassi è: Drecall"nomefile".
- **Merge:** Con questa istruzione si possono caricare in memoria degli sprite senza perdere quelli già presenti. Il caricamento avviene da nastro e la sintassi è: Merge"nomefile".
- **Dmerge:** è l'equivalente per il disco dell'istruzione precedente. La sua sintassi è Dmerge"nomefile",8.
- **Reseq:** Effettua la renumerazione degli sprite in memoria. Il primo sprite definito avrà quindi il numero 1, il secondo il numero 2 e così via. Non necessita di parametri.
- **Reserve:** Permette di incrementare l'area di memoria disponibile per gli sprite. La sintassi e Reserve nnnn dove nnnn è un intero non inferiore a 7168 (il numero minimo di byte a dispsizione degli sprite).
- **Init:** Inizializza il sistema. questa istruzione dovrebbe sempre essere posta all'inizio di ogni programma che utilizza la grafica.
- **Hires:** Pone il computer nel modo alta risoluzione.
- **Lores:** Riporta il computer al modo testo.
- **Eback:** Attiva il modo extended background.
- **Costanti di colori:** Tutti i colori disponibili sul C64 sono identificati da costanti predefinite. Eccole: Black, White, Red, Cyan, Purple, Green, Blue, Yellow, Orange, Brown, .Red, Gray1, Gray2, .Green, .Blue, Gray3. Le costanti precedute dal carattere "." (punto) corrispondono ai colori accesi (light).
- **Tborder:** Cambia il colore del bordo dello schermo (in modo testo). La sintassi è Tborder colore, dove colore è un intero compreso fra 0 e 15 oppure una delle costanti predefinite viste prima.
- **Hborder:** Cambia il colore del bordo in alta risoluzione. La sintassi è identica a quella dell'istruzione precedente.
- **Tpaper:** Cambia il colore dello sfondo dello schermo in bassa risoluzione. La sintassi è Tpaper colore.

• **Hpaper:** Cambia il colore dello sfondo dello schermo in alta risoluzione.
• **Bg0, Bg1, Bg2, Bg3:** con queste istruzioni si possono settare i 4 colori del modo extended background. La sintassi è identica a quella delle ultime istruzioni viste.
• **Ink:** Setta il colore di visualizzazione dei caratteri in modo testo.
• **Multi:** Predispone il modo multicolore, sia per la bassa che per l'alta risoluzione. Cioè Lores:Multi e Hires:Multi predispongono il computer per il modo testo multicolore e per l'alta risoluzione multicolore.
• **Mono:** Fa ritornare al modo monocromatico ed è utilizzabile sia per l'alta che per la bassa risoluzione.
• **Window:** Consente di avere contemporaneamente testo e alta risoluzione. La sintassi è Window n dove n è il numero di linee che, a partire dall'alto, devono essere predisposte per l'alta risoluzione.
• **Atton:** Attiva gli attributi principali degli sprite. Non necessita di alcun parametro.
• **Attof:** Disattiva gli attributi principali. Anche in questo caso non è necessario specificare alcun parametro.
• **Att2on:** Attiva entrambi gli insiemi di attributi principali e secondari. Non si deve specificare nessun parametro.
• **S2col:** Predispone il modo grafico in due colori. Dopo che viene eseguito questo comando ogni istruzione grafica che agisce sui pixel (punti) funziona tenendo presente che la visualizzazione è in 2 colori.
• **S4col:** Predispone il modo grafico per 4 colori. Anche in questo caso l'effetto del comando si fa sentire sulle istruzioni che accedono ai pixel, secondo i principi visti per il comando precedente.
• **Fgnd:** Setta il colore di linea per l'alta risoluzione. La sintassi è Fgnd n, dove n è il codice di un colore.
• **Bgnd:** Setta il colore di sfondo per l'alta risoluzione. La sintassi è identica a quella dell'istruzione precedente.
• **Mcol1:** Fissa il colore 1 per l'alta risoluzione multicolore. La sintassi è Mcol1 n, dove n è il codice di un colore.
• **Mcol2:** Setta il colore 2 per l'alta risoluzione.
• **Mcol3:** Setta il colore 3 per l'alta risoluzione.

Le ultime 5 istruzioni viste agiscono sulla variabile Atr, cioè la variabile che contiene i valori degli attributi principale e secondario.

• **Seta:** Setta gli attributi principale e secondario di una parte di uno sprite. La sintassi è Seta Spn, x, y, incx, incy, Atr. Spn è il numero dello sprite su cui si vuole agire. X e y sono le coordinate dell'angolo superiore sinistro della parte di sprite di cui si vogliono modificare gli attributi. Incx è la larghezza e incy l'altezza della porzione di sprite. Atr è la variabile che contiene gli attributi.
• **Switch:** Fa scambiare l'attributo principale con quello secondario di ogni sprite.
• **Norm:** Ripristina gli attributi principale e secondario di ogni sprite al loro valore originario (quindi questo comando può essere utilizzato per riportare la situazione alla normalità dopo che si è impiegato il comando Switch).
• **Wclr:** Questa istruzione serve per cancellare una parte di un hires sprite. La sintassi è Wclr spn, x, y, incx, incy, Atr. Spn è il codice dello sprite su cui si vuole agire. X e y sono le coordinate dell'angolo in alto a sinistra della porzione di sprite da cancellare. Incx e incy sono la larghezza e l'altezza della porzione di sprite da cancellare. Atr è la variabile degli attributi. Wclr cancella la parte selezionata dell'hires sprite cambiando contemporaneamente i colori di sfondo e visualizzazione. I nuovi colori sono contenuti nella variabile sprite Atr e possono essere modificati con le istruzioni Bgnd, Fgnd, Mcol.
• **Sclr:** Cancella completamente un Hi-Res sprite. La sintassi è Sclr spn, Atr. Il ruolo di spn e Atr è lo stesso visto per l'istruzione precedente.
• **Mode:** Setta il colore di visualizzazione per le istruzioni grafiche. a sintassi è Mode n, dove n deve essere un intero fra 0. Se n vale 4 allora i pixel a cui si accede con le istruzioni grafiche vengono invertiti, sia nel caso in cui si stia agendo su un Hi-Res sprite monocromatico che su uno multicolor. Se n vale 0 o 1 allora i pixel a cui si accede vengono cancellati (cioè posti nello stesso colore di sfondo) se si sta agendo su un Hi-Res sprite monocromatico mentre vengono posti nel colore 0 o 1 se si sta agendo su un Hi-Res sprite multicolor. Se n vale 2 o 3 i pixel a cui si accede vengono posti nel colore di visualizzazione nel caso in cui si sta agendo su un Hi-Res sprite momocromatico mentre vengono posti nei colori 2 e 3 se lo sprite è multicolor.
• **Plot:** Consente di settare un pixel. La sintassi è Plot spn, x, y. Spn è il numero dello sprite su cui si vuole agire. X e y sono le coordinate del punto da manipolare (secondo la modalità indicata mediante l'istruzione Mode).
• **Box:** Traccia un rettangolo. La sintassi è Block spn, x, y, incx, incy. X e y sono le coordinate dell'angolo in alto a sinistra del rettangolo da visualizzare nello sprite spn. Incx e incy sono la larghezza e l'altezza del rettangolo.
• **Draw:** Traccia una linea. La sintassi è Draw spn, x1, y1, x2, y2. X1, y1, x2 e y2 sono le coordinate dei punti fra cui verrà tracciata la linea.
• **Poly:** Traccia un poligono. La sintassi è Poly spn, x, y, incx, incy, n, scl (vedi **figura 1**).
• **Fill:** Riempie un'area chiusa. La sintassi è spn, x, y, n. X e y sono le coordinate di un punto all'interno dell'area chiusa da riempire. N ha la stessa funzione vista per l'istruzione Mode.
• **Point:** Indica lo stato di un punto. La sintassi è Point (spn, x, y). X e y sono le coordinate del punto da testare. Il risultato di questa funzione è un intero fra 0 e 4 che corrisponde al colore (vedere istruzione Mode) del punto.
• **Putblk:** Visualizza uno sprite sulla pagina in alta risoluzione. La sintassi è Putblk spn, x, y. X e y sono le coordinate dell'angolo in alto a sinistra dello sprite, collocato all'interno della pagina in alta risoluzione. Spn è lo sprite da copiare in alta risoluzione.
• **Putor:** Copia uno sprite nella pagina in alta risoluzione eseguendo un Or fra i punti dello sprite e quelli della pagina in alta risoluzione. In pratica questa istruzione, a differenza di Putblk, mantiene inalterati i punti eventualmente settati della pagina in alta risoluzione. La sintassi è identica a quella di Putblk.
• **Putand:** Copia uno sprite nella pagina grafica eseguendo un And fra i punti dello sprite e quelli della pagina grafica. La sintassi è identica a quella di Putblk.
• **Putxor:** Copia uno sprite nella pagina grafica eseguendo un Xor (or esclusivo) fra i punti dello sprite e quelli della pagina grafica.
• **Getblk:** Esegue l'operazione inversa di Putblk e quindi copia una parte della pagina grafica in uno sprite. La sintassi è Getblk spn, x, y. Spn è lo sprite su cui verrà copiata la porzione di pagina in alta risoluzione. La parte di pagina grafica da copiare è quella che va dal punto di coordinate x e y sino alla fine dello schermo.
• **Getor:** Esegue l'operazione contraria di Putor. La sintassi è identica a quella di Getblk.
• **Getxor:** Esegue l'operazione contraria di Putxor. La sintassi è la stessa di quella di Getblk.
• **Movblk:** Copia una parte di uno sprite in un altro sprite. La sintassi è Movblk spn1, x1, y1, incx, incy, spn2, x2, y2. La por-

zione dello sprite spn1 definita da x, y, incx e incy viene copiata nello sprite spn2 a partire dal punto di coordinate x2, y2

- **Movor:** Esegue la stessa operazione svolta da Movblk e in più esegue un Or fra lo sprite sorgente e quello destinazione. La sintassi è identica a quella vista per Movblk.
- **Movand:** Esegue la stessa operazione svolta da Movblk eseguendo in più un And fra lo sprite sorgente e quello destinazione. La sintassi è identica a quella vista per Movblk.
- **Movxor:** Esegue la stessa operazione svolta da Movblk eseguendo in più un Or esclusivo fra lo sprite sorgente e quello destinazione. La sintassi è identica a quella vista per Movblk.
- **Cpyblk:** Copia uno sprite in un altro. La sintassi è Cpyblk spn1, spn2. Lo sprite spn1 viene copiato nello sprite spn2.
- **Cpyor:** Esegue la stessa operazione svolta da Cpyblk eseguendo in più un Or fra lo sprite sorgente e quello destinazione. La sintassi è identica a quella vista per Cpyblk.
- **Cpyand:** Esegue la stessa operazione svolta da Cpyblk eseguendo in più un And fra lo sprite sorgente e quello destinazione. La sintassi è identica a quella vista per Cpyblk.
- **Cpyxor:** Esegue la stessa operazione svolta da Cpyblk eseguendo in più un Or esclusivo fra lo sprite sorgente e quello destinazione. La sintassi è identica a quella vista per Cpyblk.

Le seguenti 16 istruzioni permettono di scambiare fra loro porzioni di sprite. Piuttosto che spiegare una per una ciascuna istruzione proponiamo un breve demo che ne mostra molto chiaramente il funzionamento.

- **Blk%blk, Blk%or, Blk%xor, Blk%and, Or%blk, Or%or, Or%xor, Or%and, Xor%blk, Xor%or, Xor%xor, Xor%and, And%blk, And%or, And%xor, And%and.**

```
10 Init: Fgnd Green: Bgnd White: Window 8
20 For n=0 to 15
30 if n:- Repeat: Get a$: Until a$<>""
40 Print chr$(147)
50 Sclr 0, Atr: Window 8
60 For i=0 to 48 step 16
70 For j=0 to 48 step 16
80 If i+j And 16 Box 0,i,j,16,16
90 Next j,i
100 For i=0 to 6 step 2
110 Putblk 40,8,i:Next
120 For j=0 to 400:next
130 Spn=0:Col=0:Row=0:Wid=8:Hgt=8
140 Spn2=0:Col2=8:Row2=0
150 Case n
170 Of 0:blk%blk
180 Of 1:blk%or
190 Of 2:blk%xor
200 Of 3:blk%and
210 Of 4:or%blk
220 Of 5:or%or
230 Of 6:or%xor
240 Of 7:or%and
250 Of 8:xor%blk
260 Of 9:xor%or
270 Of 10:xor%xor
280 Of 11:xor%and
290 Of 12:and%blk
300 Of 13:and%or
310 Of 14:and%xor
315 Of 15:and%and
320 Casend
330 next n
```

- **Dtcton:** Attiva la rilevazione di collisioni fra sprite.
- **Dtctoff:** Disattiva la rilevazione di collisioni fra sprite. Quando è attiva la rilevazione delle collisioni se due sprite entrano in contatto le due variabili sprite Ccol e Crow contengono la riga e la colonna in cui si è verificata la collisione. Se non si è verificata una collisione entrambe le variabili valgono -1.
- **Flip:** Capovolge una parte di uno sprite (cioè calcola la sua immagine speculare rispetto alla linea orizzontale). La sintassi è Flip spn, x, y, incx, incy. La porzione dello sprite spn selezionata viene capovolta rispetto all'orizzontale.
- **Flipa:** Esegue la stessa operazione dell'istruzione precedente spostando oltre ai pixel anche i colori.
- **Mir:** Calcola l'immagine speculare di una porzione di uno sprite rispetto alla verticale. Ha la stessa sintassi di Flip.
- **Mar:** Esegue la stessa operazione di Mir spostando oltre ai pixel anche i colori.
- **Exx:** Espande in direzione orizzontale una porzione di uno sprite. La sintassi è Exx spn1, x1, y1, incx, incy, spn2, x2, y2. La porzione selezionata dello sprite spn1 viene copiata, ingrandita secondo un fattore pari a 2, nello sprite spn2 a partire dalla posizione di coordinate x2, y2.
- **Exy:** Espande in direzione verticale una parte di uno sprite. La sintassi è identica a Exx.
- **Spin:** Ruota di 90 gradi una parte di uno sprite. La sintassi è identica a quella dell'istruzione Exx.
- **Inv:** Inverte tutti i pixel di una parte di uno sprite. La sintassi è Inv spn, x, y, incx, incy.
- **Movatt:** Muove gli attributi principale e secondario di una parte di uno sprite in un altro sprite. La sintassi è Movatt spn1, x, y, incx, incy, spn2, x2, y2. Gli attributi della porzione selezionata dello sprite spn1 verranno copiati nello sprite spn2.
- **Swaptt:** Scambia gli attributi di due sprite. La sintassi è identica a quella dell'istruzione precedente.
- **Scan:** Permette di ispezionare una parte di uno sprite per verificare se tutti i pixel sono visibili (cioè se sono di colore diverso dallo sfondo). La sintassi è Scan spn, x, y, incx, incy. Se tutti i pixel della porzione di sprite selezionata sono del colore di sfondo questa funzione ritorna il valore 0 (o false). Diversamente ritorna il valore -1 (o true).
- **Attget:** Permette di ispezionare gli attributi di una parte di uno sprite. La sintassi è Attget spn, x, y. Se prima di questa istruzione è stato invocato il comando Atton verrà letto solo l'attributo principale; se è stato invocato il comando Att2on allora verranno letti sia l'attributo principale che quello secondario. I valori letti vengono posti nella variabile sprite Atr.

Le seguenti 12 istruzioni permettono di scrollare una porzione di uno sprite.

- **Scr1, Scr2, Scr3, Scl1, Scl2, Scl3, Wrr1, Wrr2, Wrr3, Wrl1, Wrl2, Wrl3.** Il numero che compare al termine di ogni istruzione indica il numero di pixel di cui lo sprite sarà scrollato. La direzione delo scrolling è determinata dall'ultima lettera: se è una r allora lo scrolling è verso destra mentre se è una l avverrà verso sinistra. La prima lettera indica se lo scrolling avviene con o senza wrap round (quando lo scrolling viene effettuato con wrap round tutti i pixel che escono dalla porzione di sprite da scrollare ricompaiono sul lato opposto dello sprite). Se questa lettera è una s lo scrolling è senza wrap round mentre se è una w lo scrolling è con wrap round. La sintassi è la stessa per tutte le istruzioni per cui viene indicata solo per una di esse.
- **Scr1 spn, x, y, incx, incy.** In questo caso la porzione selezionata dello sprite spn viene scrollata verso destra di un 1 pixel.

• **Rpt:** Ripete un comando di scrolling. La sintassi è Rpt n dove n è il numero di volte che il comando di scrolling deve essere ripetuto. Ovviamente questa istruzione deve essere seguita da uno dei comandi di scrolling visti precedentemente.
• **Raster:** Permette di selezionare la parte di schermo su cui avrà effetto una istruzione grafica.
La sintassi è Raster n dove n è la linea del video a partire dalla quale ogni istruzione grafica può agire. Questa istruzione non può esesre utilizzata con le istruzioni di scrolling. In questo caso si deve utilizzare l'istruzione Rsync, che ha la stessa sintassi di Raster.
• **Wrap:** Effettua lo scrolling verticale di una porzione di sprite con il wrap round. La sintassi è Wrap spn, x, y, incx, incy, n. La parte selezionata dello sprite viene scrollata verticalmente di n pixel.
• **Scroll:** Effettua lo scrolling verticale di una porzione di sprite senza wrap round. Anche in questo caso la sintassi è identica a quella dell'istruzione precedente.

Le 4 istruzioni che seguono consentono di effettuare lo scrolling verticale e orizzontale degli attributi principale e secondario di parte di uno sprite. Lo scrolling è di 8 pixel per volta.
• **Att, Attr, Attup, Attdn.** Le prime due istruzioni effettuano lo scrolling in direzione orizzontale rispettivamente verso sinistra e destra. Le ultime due istruzioni effettuano lo scrolling in direzione verticale rispettivamente verso l'alto e verso il basso. La sintassi è la stessa per tutte le istruzioni quindi diamo solo quella di una di esse. Attl spn, x, y, incx, incy.
• **Lcase:** Riporta al set minuscolo.
• **Ucase:** Riporta al set maiuscolo.
• **Char:** Pone un carattere all'interno di uno sprite. La sintassi è Char spn, x, y, n. Il parametro n è il codice ascii del carattere da mettere nello sprite mentre x e y rappresentano le coordinate alle quali deve essere messo. Di seguito riportiamo l'insieme di valori che può assumere il parametro n.
Da 0 a 255 caratteri normali. Da 256 a 511 caratteri in reverse. Da 512 a 767 caratteri normali. Da 1024 a 1279 caratteri normali in dimensione doppia. Da 1280 a 1535 caratteri in reverse in dimensione doppia. Da 1536 a 1791 caratteri normali in dimensione doppia.
• **Text:** Pone una stringa all'interno di uno sprite. La sintassi è Text spn x, y, stringa, offset. X e y sono le coordinate alle quali verrà posta la stringa.
Offset indica il valore numerico da aggiungere al codice ascii di ogni carattere presente nella stringa. Utilizzando questo parametro si possono selezionare tutti i set di caratteri disponibili (normali, reverse e in dimensione doppia).
• **Putchar:** Questa istruzione permete di ridefinire un carattere utilizzando una parte, delle dimensioni di 8x8 punti, di uno sprite. La sintassi è Putchar spn, x, y, n. X e y sono le coordinate dell'angolo superiore sinistro del blocco di 8x8 pixel dello sprite da utilizzare nella ridefinizione del carattere. N è il codice ascii del carattere da ridefinire.
• **Conv:** Prende una porzione, delle dimensioni di 24x21 pixel, di un software sprite e lo pone nell'area di definizione di un hardware sprite.
Per memorizzare i dati di un hardware sprite si hanno a disposizione 32 blocchi di memoria ciascuno dei quali può contenere la definizione di un hardware sprite. Questi blocchi sono numerati da 0 a 31.
I blocchi che vanno da 0 a 15 sono utilizzati anche per memorizzare i dati del set di caratteri. La sintassi dell'istruzione è Conv spn, x, y, bl. La porzione di 24x21 pixel viene prelevata a partire dal punto di coordinate x, y e viene posta nel blocco bl dell'area riservata agli hardware sprite.
• **Hset:** Associa a uno sprite un blocco di dati. La sintassi è Hset n, bl. Il parametro n deve esere un intero fra 0 e 7 e indica il codice dell'hardware sprite mentre n è il blocco di dati che contiene la sua definizione.
• **Hcol:** Definisce il colore di un hardware sprite. La sintassi è Hcol n, col. Allo sprite n viene associato il colore col.
• **Hon:** Rende visibile un hardware sprite. La sintassi è Hon n. Lo sprite n viene reso visibile.
• **Hoff:** Disattiva uno sprite. La sintassi è Hoff n. Lo sprite n viene disabilitato.
• **Hexx:** Espande un hardware sprite in direzione orizzontale. La sintassi è Hexx n. n è lo sprite da espandere.
• **Hexy:** Espande un hardware sprite in direzione verticale. La sintassi è Hexy n. N è lo sprite da espandere.
• **Hshx:** Riporta alla normalità la dimensione orizzontale di un hardware sprite. La sintassi è Hshx. N è o sprite da ridurre.
• **Hshy:** Riporta alla normalità la dimensione verticale di un hardware sprite. La sintassi è Hshy n. N è lo sprite da ridurre.
• **Over:** Setta la priorità di visualizzazione di un hardware sprite. La sintassi è Over n. I caratteri avranno la priorità sullo sprite n.
• **Under:** Setta la priorità di visualizzazione di un hardware sprite. La sintassi è Under n. Lo sprite n avrà la priorità sui caratteri.
• **Move:** Muove automaticamente un hardware sprite. Il movimento viene gestito da interrupt. La sintassi è Move n, incx, incy. Le coordinate dello sprite n verranno costantemente incrementate di incx e incy unità.
• **Hy:** Ritorna l'ordinata dell'hardware sprite n. La sintassi è Hy (n) dove n è lo sprite di cui si vuole conoscere il valore della coordinata verticale.
• **Hx:** Ritorna il valore dell'ascissa di un hardware sprite. La sintassi è identica a quella di Hy.
• **Track:** Muove automaticamente un harware sprite. Questa istruzione come Move muove permette di muovere uno sprite automaticamente. Inoltre modifica gli incrementi dell'ascissa e dell'ordinata dell'harware sprite ogni volta che questo raggiunde il limite dello schermo. I dati vengono letti da un software sprite. La sintassi è Track hsp, ssp. I dati che definiscono il software sprite ssp verranno letti a coppie e utilizzati rispettivamente come nuovo incremento dell'ascissa e dell'ordinata dell'harware sprite hsp. Track può essere utilizzata anche come funzione. In questo caso permette di stabilire se un hardware sprite è in movimento. La sintassi diventa allora Track (n).
• **H2col:** Setta il modo monocromatico per un hardware sprite. La sintassi è H2col n, dove n è lo sprite che deve essere visualizzato in due colori.
• **H4col:** Setta il modo multicolore per un hardware sprite. La sintassi è identica a quella dell'istruzione precedente.
• **H1col:** Setta il colore 1 per gli hardware sprite multicolor. La sintassi è H1col c, dove c è il codice di un colore.
• **H3col:** Setta il colore 3 per gli hardware sprite multicolor. La sintassi è identica a quella dell'istruzione precedente.
Le due istruzioni precedenti servono a definire i colori comuni a tutti gli sprite multicolor utilizzati. Gli altri 2 colori utilizzabili per ogni sprite vengono definiti indipendentemente l'uno dall'altro con le istruzioni già viste per il modo monocromatico.
• **Hit:** Testa le collisioni degli hardware sprite. La sintassi è Hit (n, cod), dove n è l'harware sprite che si vuole tenere sotto controllo mentre cod è il codice relativo al tipo di collisione che si

vuole intercettare. Il valore da utilizzare per questo parametro è 0 se si vuole intercettare ogni collisione fra lo sprite specificato e un altro sprite; 8 per le collisioni fra lo sprite indicato e i caratteri. Con questi valori ogni collisione rilevata cancella dalla memoria ogni precedente collisione. Per evitare che questo succeda si devono utilizzare i valori 16 e 24, rispettivamente per le collisioni sprite-sprite e sprite-carattere. Il valore ritornato dalla funzione è false, 0, se non si è verificata alcuna collisione e true, -1, se si è verificata.
• **Dblank:** Attiva lo screen blanking.
• **Dshow:** Disattiva lo screen blanking.
• **H38col:** Attiva la visualizzazione su 38 colonne e 24 linee. Questo comando dovrebbe sempre essere utilizzato quando si effettua lo scrolling orizzontale o verticale dello schermo.
• **H40col:** Riporta la visualizzazione su 40 colonne e 25 linee.
• **Scrx:** Scrolla lo schermo in direzione orizzontale. La sintassi è Scrx n, dove n è il numero di pixel di cui deve essere scrollato lo schermo.
• **Scry:** Scrolla lo schermo in direzione verticale. La sintassi è Scry n, dove n è il numero di pixel di cui deve essere scrollato lo schermo.

Suono

• **Sidclr:** Inizializza il chip che gestisce gli effetti sonori. Questo comando dovrebbe sempre essere posto all'inizio dei programmi che fanno uso di effetti sonori.
• **Volume:** Setta il volume. La sintassi è Volume n, dove n è il valore del volume.
• **Frq:** Setta la frequenza di una voce. La sintassi è Frq v, f. V è la voce mentre f è la frequenza.
• **Adsr:** Setta i parametri attack, decay, sustain e release per una voce. La sintassi è Adsr v, attack, decay, sustain, release. V è la voce su cui si vuole agire.
• **Saw:** Attiva l'onda a dente di sega per una voce. La sintassi è Saw v, dove v è la voce.
• **Tri:** Attiva l'onda triangolare per una voce. La sintassi è Tri v, dove v è la voce.
• **Pulse:** Attiva l'onda quadra per una voce. La sintassi è Pulse v, dove v è la voce.
• **Noise:** Attiva il rumore bianco per una voce. La sintassi è Noise v, dove v è la voce.
• **Music:** Attiva una voce. La sintassi è Music v, t. Il parametro v indica la voce da attivare mentre t è la lunghezza della nota da suonare.
• **Play:** Attiva le voci da interruzione. Con questa istruzione è possibile suonare delle note da interruzione. I dati relativi alle note da suonare sono letti dai software sprite. La sintassi è Play v1, v2, v3. I tre parametri sono i software sprite da cui verranno prelevati i codici delle note da suonare, rispettivamente, con la voce 1, 2 e 3.
• **Rplay:** Attiva le voci da interruzione. Come l'istruzione precedente Rplay permette di attivare le voci da interruzione. Inoltre ripete automaticamente la lettura dei dati quando questi sono stati completamente letti.
• **Cutoff:** Setta la frequenza di Cutoff utilizzata dal filtro del Sid. La sintassi è Cutoff f, dove f è la frequenza da assegnare al filtro.
• **Pass:** Seleziona il filtro. La sintassi è Pass n, dove n è il codice numerico del filtro che si vuole selzionare. I codici dei filtri disponibili sono: 0 per filtro low pass; 1 per il filtro high pass; 2 per il filtro band pass; 3 per il filtro notch reject.
• **Resonance:** Attiva la risonanza. La sintassi è Resonance n, dove n deve essere un intero fra 0 e 15 che indica il valore della risonanza.
• **Filter:** Attiva il filtro di una voce. La sintassi è Filter v, True o False. Il parametro v è relativo alla voce di cui si vuole attivare il filtro mentre il parametro successivo deve essere True per attivare il filtro e False per disattivarlo.
• **Ring:** Attiva la modulazione ad anello (ring modulation). La sintassi è Ring v, True (o False). Il parametro v indica la voce che deve esseremodulata. La modulazione avviene fra la voce selezionata e la voce 3.
Il secondo parametro indica se la modulazione deve essere attivata, True, o disattivata, False.
• **Sync:** Attiva la sincronizzazione. La sintassi è Sync v, True o False. Il ruolo dei parametri è identico a quello visto per l'istruzione Ring e anche in questo caso la sincronizzazione avviene con la voce 3.
• **Mute:** Disattiva la voce 3.
• **Osc:** Ritorna il valore dell'oscillatore della voce 3.
• **Env:** Ritorna il valore del generatore di envelope.

Multi tasking

Laser Basic permette di eseguire contemporaneamente fino a 3 programmi in memoria, permettendo anche la comunicazione fra di essi. La prima cosa da fare per definire i programmi da eseguire in multitasking è allocare la memoria necessaria per le variabili di ciascun programma o task. L'istruzione necessaria è Task m1, m2.I parametri m1 e m2 indicano il numero di byte disponibili per le variabili dei task 1 e 2. Il terzo task cioè il task 0 viene gestito automaticamente dal computer e quindi per questo non è necessario allocare la memoria. Per lanciare un task si deve utilizzare l'istruzione Task n, numlinea o label. Il primo parametro n indica il task da eseguire (tenete presente che il task 0 viene sempre eseguito automaticamente) mentre il parametro successivo indica il numero di linea (oppure la label ad esso associata) da cui inizia il task. Per fermare l'istruzione di un Task si può utilizzare l'istruzione Halt n, dove n è il task da interrompere. Se non viene specificato alcun parametro verrà interrotto il task corrente.

Tenete presente che gli eventuali errori nei task verranno visualizzati preceduti dal numero che identifica il task in cui sono stati rilevati.

Istruzioni di uso generale

• **Kb:** Questa funzione, la cui sintassi è Kb(c), ritorna il valore True se è premuto il tasto di codice c, mentre ritorna il valore False se tale tasto non viene premuto. I codici dei caratteri utilizzabili con questa funzione sono diversi dai codici ascii.
• **Fire1:** Ritorna il valore True se è stato premuto il tasto di fire del joystick in porta 1 (False in caso contrario).
Fire2 ritorna il valore True se è stato premuto il tasto di fire del joystick in porta 2 (False in caso contrario).
• **Js1:** Ritorna la direzione di movimento impressa alla leva del joystick in porta 1. I valori che si possono ottenere sono: 0 nel caso in cui la leva non è stata mossa; 1, 2, 3, 4, 5, 6, 7, 8 se la leva del joystick è stata mossa in direzione ore 3, 2, 12, 10, 9, 8, 6, 4 rispettivamente.
• **Js2:** Come sopra con il joystick in porta 2.
• **Lpx:** Ritorna la coordinata orizzontale dell'impulso della penna ottica.
• **Lpy:** Ritorna la coordinata verticale dell'impulso della penna ottica.

Silvia Alessi e
ndrea Rebosio

Sprite Generator

Uno strumento fondamentale per realizzare tutte le immagini per Laser Basic. Disponibili tre eccezionali librerie di figure.

Questo programma è stato sviluppato per facilitare la progettazione dei software sprite da utilizzarsi poi con il Laser Basic. Quest'ultimo infatti ha tutte le opzioni per gestire il movimento e la collocazione sullo schermo degli sprite, tuttavia non possiede al suo interno alcuna facility per la realizzazione. Ciò vuol dire che la realizzazione dei programmi e dei giochi con Laser Basic è divisa in due momenti ben distinti: la progettazione delle immagini e l'implementazione delle istruzioni che li utilizzano. La prima fase va appunto effettuata con il programma Sprite Generator di cui adesso parleremo. Va detto che per chi non ha inclinazioni artistiche e che quindi ha difficoltà nel tradurre in pratica quello che ha in mente a livello di immagini, è disponibile una completissima libreria di immagini tra quelle più frequentemente utilizzabili. Per caricare Sprite Generator si può procedere direttamente dal menù di TuttoCommodore oppure caricandolo dalla directory del dischetto (lato 1) con l'istruzione LOAD"SPTGEN",8 e dando al termine RUN.

Nel corso della trattazione verranno usate alcune notazioni simboliche: SQR indicherà la griglia di 8x8 (8x4 in multicolor) di definizione degli sprite in alto sulla sinistra dello schermo. CHR$ indicherà il cursore relativo alla griglia SQR. HiresScreen indica invece l'area di 15x30 caratteri sulla quale viene creato l'intero software sprite. Con Ink e Paper verranno indicati i 3 colori di linea più quello dello sfondo.

L'area di schermo con cui si sta lavorando e definita dai parametri COL, ROW, HGT e LEN, può essere evidenziata premendo la barra spaziatrice.

Ci sono 5 modi operativi per utilizzare il generatore di sprite. Alcune funzioni sono disponibili in più di un modo. Per selezionare un modo operativo è necessario premere il tasto Commodore (in basso a sinistra) insieme con un numero da 1 a 5. Lasciate il tasto Commodore una volta che siete entrati nel modo desiderato. Per uscire premete nuovamente il tasto Commodore.

Modo 1

Consente di editare la griglia SQR con due colori. Una volta entrati in questo modo verrà visualizzato il cursore CHR$ lampeggiante.

• I tasti cursore sono utilizzati per spostare CHR$ all'interno di SQR (64 celle). Utilizzate lo shift destro e non quello sinistro per spostarvi a sinistra e in alto. Per accendere la cella su vi trovate premete il tasto "3". La cella corrispondente diventerà nera. Per spegnere la cella premete il tasto "4". Questi sono gli unici comandi per editare la griglia SQR.

• I tasti D e U. Questi due comandi consentono rispettivamente di scrivere o leggere la griglia SQR dall'HiresScreen partendo dalla posizione corrente, ovvero quella indicata nel pannello di controllo dalle variabili COL e ROW. Inoltre la posizione corrente può essere evidenziata in lampeggio premendo la barra spaziatrice. Una volta che la porzione Hires del disegno è stata caricata in SQR con il tasto U può essere modificata (vedremo poi come) e quindi risalvata in HiresScreen con il tasto D. Questo modo di procedere consente di limitare gli errori e consentendo di provare a disegnare senza compromettere il disegno già fatto. Se la variabile ATTR del pannello di controllo vale zero non verranno caricati gli attributi colore dell'Hires Screen. Viceversa se ATTR vale uno tutti gli attributi verranno impostati. Questo discorso vale nei due sensi, cioè sia in caricamento che in salvataggio su HiresScreen.

• Premendo il tasto A viene cambiato il valore della variabile ATTR, passando alternativamente da 0 a 1. ATTR in generale controlla l'effetto dei comandi sugli attributi colore: se vale 0 il colore non viene coinvolto, se vale 1, sì.

• Il tasto E viene utilizzato per modificare i valori di Ink e Paper. Viene richiesto il numero di Ink che si vuole cambiare (da 1 a 3 indicano i tre diversi INK, 4 indica Paper) anche se nel modo 1 si dispongono solo di 2 colori, Ink 1 e Paper. Quindi occorre indicare il nuovo valore dell'Ink selezionato digitando un numero da 0 a 15.

• Il tasto N consente di inserire i dati per il disegno di SQR utilizzando la codifica in byte della griglia come quando si ridefiniscono i caratteri. Un prompt chiederà via via i valori decimali degli 8 byte necessari. La codifica rispecchia la struttura binaria dell'immagine: 1=on 0=off per ogni pixel, dall'alto in basso e con il bit più significativo a sinistra.

• Il tasto C cancella il quadrato CHR$.

• La barra spaziatrice consente di visualizzare la finestra di alta risoluzione e in congiunzione con i tasti di movimento cursore consente di spostarla all'interno dell'HiresScreen.

Premendo il tasto Commodore si ritorna al modo 0 e il programma si mette in attesa del nuovo modo operativo.

Utilizzate il tasto shift di sinistra insieme ai tasti di movimento cursore per modificare le dimensioni della finestra:

Left Shift e CRSR Right - Allarga di un carattere.
Left Shift e CRSR Left - Restringe di un carattere.
Left Shift e CRSR Down - Alza di un carattere.
Left Shift e CRSR Up - Abbassa di un carattere.

Il pannello è aggiornato di conseguenza a queste modifiche.

• Il tasto "-" (simbolo di sottrazione) imposterà i valori di Ink e Paper correnti con quelli di HiresScreen.

• Il tasto "+" imposterà i valori correnti di Ink e Paper nel pannello a quelli del carattere nell'angolo superiore sinistro della Hires window.

Modo 2

In questo modo si possono utilizzare 4 colori per ogni carat-

tere, lavora essenzialmente con gli stessi comandi del modo 1 eccetto che l'input numerico dei dati non è disponibile (tasto N). La griglia SQR è ridotta a sole 32 celle perché come è nodo l'aumento dei colori dimezza la risoluzione grafica. Ovviamente in questo modo sono disponibili 3 Ink diversi più il solito Paper.

Come nel modo 1 lavorano i comandi D, U, A, E, C, -, +, Barra e Commodore. Particolare attenzione va fatta nell'uso della variabile ATTR perché in multicolor il pattern dipende dai colori usati più che in monocromatico. Anche in questo caso lo Shift sinistro usato insieme ai tasti CRSR modifica la finestra Hires.

Modo 3

Questo modo lavora essenzialmente sulla finestra Hires sia in modo monocromatico sia multicolor.

• Premendo i tasti W e poi I il contenuto della finestra Hires verrà invertito (i punti on diventano off e viceversa). Nessuna operazione verrà eseguita fino a quando il tasto W non verrà rilasciato. Premendo W seguito da F si otterrà l'effetto specchio verticale sulla finestra Hires. Se ATTR è 1 anche gli attributi verranno flippati, se invece ATTR è 0 verranno flippati solo i pixel. Anche in questo caso il comando non verrà eseguito fino a quando la W non verrà rilasciata.

• Premendo W e poi M si otterrà l'effetto specchio orizzontale. ATTR viene coinvolto come nel comando precedente.

• Premendo W seguito da C verrà ripulita la finestra Hires. Se ATTR vale 1 gli attributi verranno impostati al valore corrente di INK e PAPER, altrimenti INK 3 verrà settato a nero e INK 1 e INK 2 a bianco.

• Premendo S insieme ai tasti CRSR la finestra hires verrà scrollata nelle quattro direzioni senza wrap around (quindi con perdita di dati).

• Premendo R insieme ai tasti CRSR la finestra Hires verrà scrolata nella quattro direzioni con wrap around (quindi senza perdita dei dati).

• Premendo il tasto HOME la finestra Hires verrà posizionata nell'angolo superiore sinistro (COL=10, ROW=0).

• Premendo contemporaneamente lo SHIFT destro e il tasto HOME la finestra Hires verrà cancellata e posizionata nell'angolo in alto a sinistra nell'HiresScreen. Se ATTR è 1 gli attributi dell'HiresScreen verranno impostati con quelli del pannello di controllo.

• Premendo V insieme ai CRSR Up o Down l'intero HiresScreen verrà scrollato verticalmente con wrap around, dalla riga 0 alla 24. Notate che questo significa che i caratteri possono essere memorizzati anche sotto il pannello di controllo. Se ATTR è 1 gli attributi dell'HiresScreen dalla riga 0 alla 14 verranno scrollati con wrap around. Se ATTR è 0 gli attributi colore non verranno toccati.

• I comandi E, A, Barra, Commodore, Left Shift e CRSR, -, + funzionano esattamente come nei modi 1 e 2.

Modo 4

Serve soprattutto per lavorare con lo spostamento dei dati. Molte delle opzioni, sebbene non tutte, hanno a che fare con il movimento degli sprite e dello schermo.

• Il comando G serve per convertire la screen window in uno sprite. Rilasciano il tasto G si viene invitati a inserire il numero dello sprite (da 1 a 255). Se lo sprite con quel numero esiste già verrà visualizzato un messaggio di avvertimento e così pure se tentate di dare un numero illegale. Lo sprite creato usando G avrà le dimensioni e il contenuto della screen window. Se ATTR è 0 avrà gli attributi Ink e Paper settati secondo il contenuto del pannello di controllo. Dopo la creazione i valori di SPND (la fine dello spazio per gli gli sprite) e SPRITE (l'indirizzo iniziale dello sprite creato) e FREE MEMORY verranno aggiornati sul pannello.

• Il comando P serve per portare sull'HiresScreen uno sprite precedentemente archiviato in memoria, per esempio con il comando G, nella posizione della screen window. Le dimensioni della screen window non sono modificate. Quando P viene premuto appare un prompt che chiede il numero dello sprite (da 1 a 255), anche in questo caso se si fornisce un valore illegale o l'indirizzo di uno sprite non ancora definito, si viene avvertiti da un opportuno messaggio d'errore. Se ATTR è 1 verranno posti sullo schermo anche gli attributi colore dello sprite, altrimenti solo i pixel. I dati dell'HiresScreen verranno sovrascritti.

• Il comando C consente di ridefinire le dimensioni di uno sprite già creato, per evitare di sovrascrivere i dati sullo schermo quando viene estratto.

• Il comando W consente di rimuovere uno sprite dalla memoria e lascia il suo numero disponibile per una nuova definizione con G. Dopo il comando verranno aggiornati i valori di SPND e MEMORY FREE sul pannello.

Il comando I permette di aggiornare il valore di SPRITE sul pannello di controllo con l'indirizzo iniziale dello sprite di cui è fornito il numero.

• Il comando R serve per far girare delle sequenze [illegible] di sprite consecutivi con un ritardo tra due fotogramma programmabile. Per prima cosa occorre inserire il numero del primo sprite della serie, quindi numero dell'ultimo e infine il fattore di ritardo che deve essere un numero positivo compreso tra 1 e 32767.

• Il comando M serve per effettuare delle operazioni di MOVE e necessita di numerosi parametri. Questi sono necessari per specificare le due finestre coinvolte: la finestra sorgente e la finestra destinazione. Il primo prompt invita a inserire il numero dello sprite destiazione. Questo corrisponde al numero dello sprite in cui i dati devono essere mossi. I due sucessivi riguardano rispettivamente la colonna destinazione e la riga destinazione. I tre prompt che seguono chiedono nell'ordine il numero dello sprite, il numero della colonna e della riga sorgente. Occorre ora speficicare le dimensioni delle finestre, nell'ordine larghezza e altezza.

Il prompt finale chiede il tipo di operazione che deve essere compreso tra i seguenti:

- Operazione B. Il blocco sorgente è copiato nel destinazione rimpiazzando il vecchio contenuto della finestra destinazione. Se ATTR è 1 gli attributi verranno spostati con i dati dei pixel.
- Operazione A. Il blocco sorgente verrà composto con un'operazione di AND logico con quello destinazione e il risultato verrà inserito nella finestra destinazione. ATTR settato a 1 coinvolge gli attributi colore.
- Operazione O. Come nel caso A, solo che adesso viene effettuata un'operazione di OR logico.
- Operazione E. Come nel caso A, solo che l'operazione è XOR logico, cioè un OR esclusivo.
- Operazione S. Il contenuto del blocco sorgente prima di essere copiato viene ruotato di 90 gradi in senso orario. Se lo sprite ruotato non può essere contenuto nel destinazione viene visualizzato un messaggio d'errore. ATTR ha il solito significato sui colori. In modo multicolor questo comando può produrre della garbage.
- Operazione X. Il blocco sorgente prima di essere copiato viene espanso orizzontalmente di un fattore 2. Anche in questo

caso un messaggio d'errore avverte dell'eventuale fuoriuscita dei dati dalla finestra destinazione. ATTR lavora come al solito.

- Operazione Y. Funziona come X, ma in verticale.

• Il comando U carica l'intero set di caratteri in uno sprite 16 x 16 per essere modificato. Il primo prompt è per il numero dello sprite da utilizzare, il secondo è per specificare se si vuole modificare il set maiuscolo o quello minuscolo. Questa funzione trasforma uno sprite Hires in un equivalente sprite carattere. La procedura di disegno per gli sprite carattere è:

1. Decidere quale set di caratteri si vuole e metterlo in uno sprite 16 x 16.
2. Definire un set di Hires sprite con le stesse dimensioni degli sprite carattere. Gli sprite Hires devono essere fatti interamente da blocchi di un carattere dallo sprite con il set di caratteri definito nel punto 1.
3. Utilizzare la funzione T per cambiare tutti gli Hires sprite in sprite carattere.

• La funzione T chiede come prima cosa il numero dello sprite Hires, quindi il numero dello sprite carattere. Se quest'ultimo esiste già viene visualizzato un messaggio d'errore. Il prompt successivo chiede il numero del sprite con il set di caratteri. Infine viene chiesto il modo operativo, che può essere 2, 4 o E:

- 2 significa 2 colori, l'attributo colore del carattere è il colore di foreground.
- 4 significa 4 colori, l'attributo colore del carattere è lo stesso del colore Hires n. 3
- E significa extended background, l'attributo del carattere è il foreground dell'Hires. Il colore di sfondo usato (BG0, BG1, BG2 o BG3) è è il resto ottenuto dividendo il foreground dell'Hires per 4.

• I comandi A, Bara, Commodore, Left Shift e CRSR, -, +, lavorano come in modo 1.

Modo 5

Questo modo serve per salvare e caricare sprite da disco.

• Premendo S il cursore sulla CHR$ smetterà di lampeggiare e rilasciando S verrà visualizzato un prompt che chiede il nome del file. Questo è il nome del file con cui verranno memorizzati gli sprite. Allo stesso modo opera il comando L che ovviamente serve per caricare gli sprite precedentemente salvati.

• Il comando E causerà la cancellazione del file di sprite di cui verrà fornito il nome.

• Il comando R consente di rinominare un file di sprite: prima viene chiesto il nome del file su disco e quindi il nuovo nome che a questo si vuole dare.

• Il comando I serve per inizializzare il disco
• Il comando V consente di effettuare un validate
• Il comando D consente di visualizzare la directory.

Sul dischetto sono predisposte 3 librerie di software sprite con i nomi SPRITESA, SPRITESB e SPRITESC. La seconda libreria è quella utilizzata dal programma dimistrativo del Laser Basic, le altre due contengono la maggior parte delle immagini che possono essere utili per costruire giochi e programmi grafici.

Giuseppe Brigatti

DIREZIONE GENERALE E AMMINISTRAZIONE

Gruppo Editoriale JCE

via Ferri 6 - 20092 Cinisello B. (MI)
Ufficio abbonamenti tel. 02/6120586 - 6127827
Telex: 352376 Fax: 02-6127620

Direttore Responsabile
Stefano Benvenuti

Coordinamento editoriale
Francesca Marzotto

Impaginazione elettronica
Adelio Barcella

Collaboratori
Patrizia Angelo
Giorgio Caironi
Mirko Diani
Marco Gussoni
Mario Magnani
Dolma Poli

Tutto COMMODORE è una pubblicazione Gruppo Editoriale JCE - via Ferri 6 - 20092 Cinisello Balsamo (MI). Conto Corrente Postale n.315275. Una copia L. 13.000. Arretrati: il prezzo di copertina. Abbonamento 10 numeri L. 120.000. Periodico mensile. Stampa: GEMM Grafica S.r.l Paderno Dugnano (Milano). Distribuzione esclusiva per l'Italia A.&G. Marco S.p.A. Via Fortezza 27, 20126 Milano, Tel. 02/25261, Telex 350320. Registrazione Tribunale di Milano n. 84 del 9/2/87. Pubblicità inferiore al 70%.

LA COMPILATION
DELL'ANNO

QUATTRO
SIMULAZIONI
PER COMMODORE
64 E 128

DISCO
INTERAMENTE
REGISTRATO
SUI DUE LATI